4. Dasha

Program Name: Dasha.java Input File: dasha.dat

Dasha is writing a poker game, and needs help shuffling the deck. As a first step, she needs to generate random numbers. Since Dasha likes writing all the code herself, she has decided to implement her own pseudo-random number generator (PRNG).

One of the simplest PRNGs is the linear congruential generator (LCG). An LCG is defined by four values, integers `a, b, m`, and `x_0`. The first "random" value output by the LCG is `x_0`, and subsequent values are given by the formula:

```
x_{i + 1} = (a * x_i + b) % m
```

In other words, to generate the next value of the LCG, multiply the previous value by `a`, add `b`, and take the result modulo `m`.

Dasha picked some values by hand to create her LCG. She then ran the LCG, but found that it wasn't giving all outputs between 0 and m - 1, inclusive. Can you help her figure out how many numbers her LCG doesn't generate?

For example, when `a = 1`, `b = 2`, `m = 4`, and `x_0 = 1`, the LCG outputs an infinite stream of `1, 3, 1, 3, 1, 3,` ... This means that only `2 / 4` (two out of four) values are ever seen.

Input: The first line is an integer T ($0 < T \leq 50$), the number of test cases to follow. Each test case is a single line of four single-space-separated integers, "`a b m x_0`".

0 ≤ a, b < 10^9
0 < m < 10^5
0 ≤ x_0 < m

Output: For each test case, print the number of values that are possible out of the number of total values, formatted as in the sample. Do **not** perform any simplification of the fraction.

Sample input:

```
4
1 2 4 1
3 6 10 2
1 7 11 4
3 2 17 5
```

Sample output:

```
2 / 4
1 / 10
11 / 11
16 / 17
```